A Fully Parallel VLSI-implementation of the Viterbi Decoding Algorithm

Sparsø, Jens; Jørgensen, Henrik Nordtorp; Paaske, Erik; Pedersen, Steen; Rübner-Petersen, Thomas

Published in:
Proceedings of the 15th European Solid-State Circuits Conference

Publication date:
1989

Document Version
Publisher's PDF, also known as Version of record

A Fully Parallel VLSI-implementation of the Viterbi Decoding Algorithm *

J. Sparsø[1] H. N. Jørgensen[1] E. Paaske[2] S. Pedersen[1] T. Rübner-Petersen[2]

[1] Department of Computer Science, Building 344
[2] Institute of Telecommunication, Building 343
Technical University of Denmark, DK-2800 Lyngby

Abstract

In this paper we describe the implementation of a $K = 7$, $R = 1/2$ single-chip Viterbi decoder intended to operate at 10-20 Mbit/sec. We propose a general, regular and area efficient floor-plan that is also suitable for implementation of decoders for codes with different generator polynomials or different values of K.

The Shuffle-Exchange type interconnection network is implemented by organizing the 64 processing elements to form a ring. The ring is laid out in two columns, and the interconnections between non-neighbours are routed in the channel between the columns. The interconnection network occupies 32 % of the area, and the global signals (including power) occupy a further 10 %.

A test-chip containing a pair of processing elements has been fabricated via NORCHIP (the Scandinavian CMOS IC prototype implementation service). This chip has been fully tested, and it operates correctly at speeds above 26 MHz under worst-case conditions ($V_{DD} = 4.75\ V$ and $T_A = 70\ ^0C$).

1 Introduction

Convolutional coding with Viterbi decoding has become a common technique to reduce the bit error rate on airborne digital communication links.

For high information bit-rates (greater than 5 Mbit/sec), decoders are based on fully parallel implementations of the Viterbi algorithm. VLSI is the only adequate implementation medium, but placement and routing of the processing elements is known to be a difficult and area intensive task.

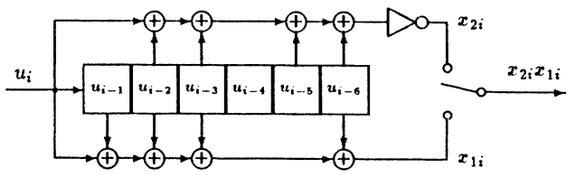
A number of specific designs have been reported [1], [2], attempts have been made to find regular and area efficient layout structures for the interconnection network [3], [4], [5] and recently a commercial $K = 7, R = 1/2$ decoder operating at 17 MHz has been announced [6].

2 The Viterbi decoding algorithm

In order to establish a notation and briefly introduce the operation of the decoder, it is convenient to start by describing the encoder. The encoder for the standardized $K = 7, R = 1/2$ code [7] is shown in figure 1. It is a synchronous state machine with 1 input u_i, 2 outputs $x_{1i}x_{2i}$ and 64 states (figure 2).

Expansion of this diagram by drawing all states for each clock cycle yields a lattice. The problem of decoding can then be expressed as finding the path through the lattice that most closely matches the received sequence as measured

*Part of this work was supported by the Danish Technical Research Council under grant no. STVF/FTU 5.17.5.6.27



Constraint length:	$K = 7$
Code rate:	$R = 1/2$
Memory depth:	$\nu = 6$ (i.e. 64 states)
Generator Polynomials:	$G1 = D^0 + D^1 + D^2 + D^3 + D^6$
	$G2 = \overline{D^0 + D^2 + D^3 + D^5 + D^6}$

Figure 1: Encoder for the $K = 7, R = 1/2$ code [7].

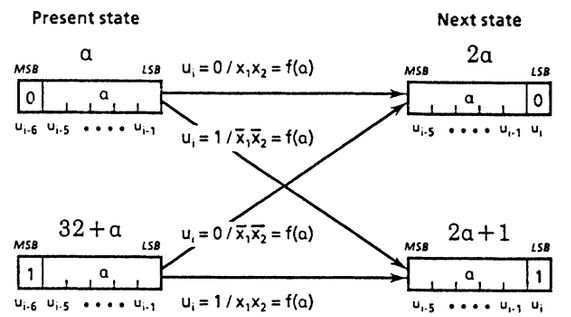


Figure 2: State graph for the $K = 7, R = 1/2$ encoder, $\alpha \in [0, 31]$. Output x_1x_2 is a function of α.

by a calculated *Path Metric* [8, chap. 6]. This involves 3 steps normally performed by 3 circuit blocks:

1. *Branch Metric Computation* – From the received bit pair (normally a pair of 3 bit "soft decisions") four **Branch Metrics** *(BM)* are calculated. They express the error assuming that the transmitted bit pair was 00, 01, 10 or 11 respectively.

2. *Path Metric Updating and Storage* – For every received bit-pair the **Path Metric** *(PM)* associated with each state is updated:

   For the two paths entering a node (state) the accumulated Path Metric is calculated by adding the BM associated with the state transition, and the PM of the preceeding node (state). The smaller of the two is stored as the new PM of the node (state).

A parallel implementation of the $K = 7$ algorithm consists of 64 such *Add-Compare-Select* processing elements (ACS-elements), and in every clock interval each of these outputs a decision.

3. *Path storage and output sequence selection* – Finally, from the 64 surviving paths – represented by the stored decisions made by the ACS-elements – the output sequence is calculated by a back-track method.

3 VLSI implementation – the wiring problem

In a fully parallel VLSI-decoder the ACS-elements and their interconnection take up the majority of the chip area, and minimizing the wiring area is an important and also difficult task. The wiring can be grouped in two categories:

Global signals that have to be distributed to all the ACS-elements. These are *Branch Metrics, clocks signals* and *power supply.*

The Path Metric interconnection network, illustrated in figure 3, for a $K = 4$ code. In this simple case the interconnection network consist of 16 PM-busses. For $K = 7$ the 64 nodes (ACS-elements) are connected by 128 PM-busses.

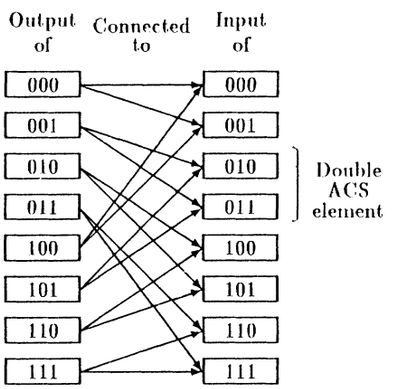


Figure 3: Path Metric interconnection network ($K = 4$).

It is a well known practice to implement the ACS-elements in pairs, as they share the same PM-inputs two by two (figure 3). This reduces the interconnection problem by a factor of two, i.e. the number of "global" PM-busses is reduced by a factor of two. Futher optimization attempts to arrange the double ACS-elements (DACS-elements) in such a way that most communication is between neighbours or near neighbours.

It appears that, for parallel implementations of the algorithm, only two systematic approaches have been reported in the literature [3], [5]: the *Shuffle-Exchange* (SE) and the *Cycle-Connected-Cubes* (CCC), of which the latter "contains even more interprocessor wiring area, but has some applications of its own" [3]. The work reported in [3] deals only with the PM-interconnection network, and the area efficiency is discussed in terms of the so-called *Thomson model* – a grid model assuming that the size of the processing elements equals the width of the PM-busses entering and leaving the element.

However, in a Viterbi decoder the processing elements are much larger than the width of the PM-busses, and preliminary results from ongoing M.Sc. thesis work show that the SE and the CCC structure has limited practical relevance for VLSI implementation of Viterbi decoders, because:

- direct implementation of the proposed interconnection graphs results in large "white areas", and because
- attempts to squeeze the layout result in an unstructured and area inefficient implementation of the interconnection graph.

4 Floor-plan

In our design we followed a more pragmatic "divide and conquer" strategy. We tried to optimize both the PM-interconnection network and the distribution of global signals, and we expressed layout-area in terms of real square-millimeters.

We found that the DACS-elements could be laid out forming one directed cycle, in which one of the two outgoing PM-busses of a DACS-element connects directely to the neighbour. By organizing the ring as two columns, routing of the remaining half of the PM-busses can be done in the channel between the columns. This structure is shown in figure 4, and a more detailed floor plan of a DACS-element is shown in figure 5. The required interconnections are derived from figure 2, and the placement of the DACS-elements shown in figure 4 is the result of an exhaustive search. This is explained in more detail below.

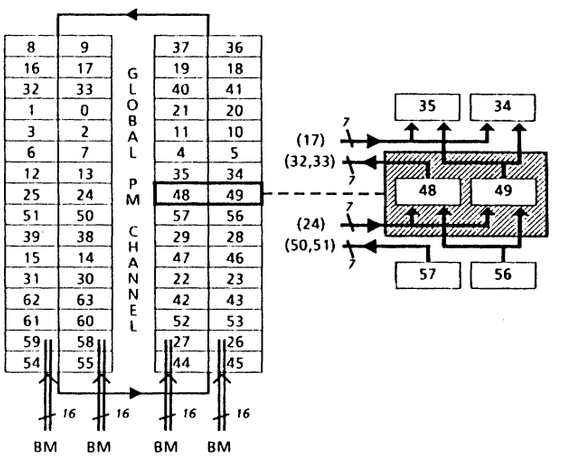


Figure 4: Floor-plan of the complete ACS-block.

The structure shown in figure 4 and figure 5 has a number of advantages and points to note:

- Routing of the PM-busses is done in one global and 32 local channels.
- The ACS-element whose output connects to the neighbour is placed outermost, allowing the two outgoing PM-busses to share a track in the local channel.
- By careful placement of the nodes, routing of the 32 PM-busses in the global channel has been reduced to 11 vertical PM-bus tracks.

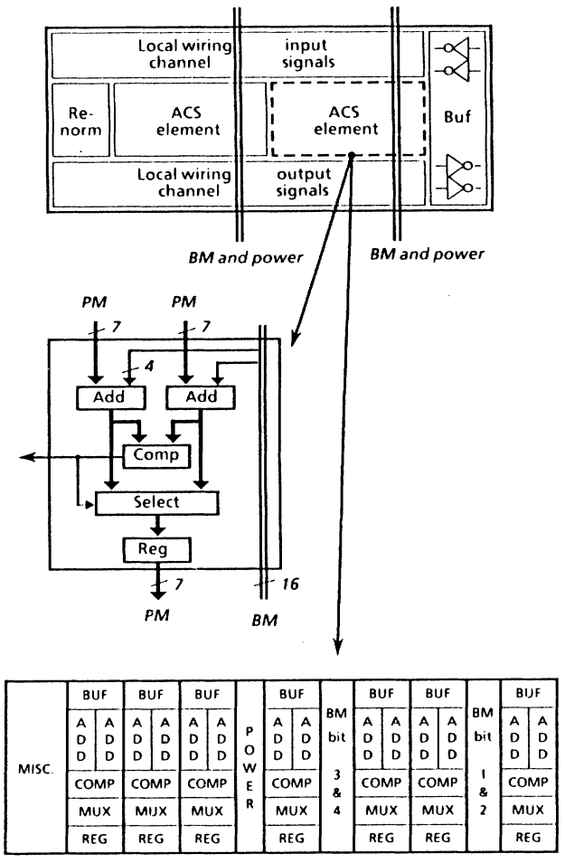


Figure 5: Floor-plan of a DACS-element.

- The 64 decision signals are available at the periphery of the structure. Also, the global clock signals connect here.
- The BM-busses and power-rails run vertically, through each of the four columns of (single) ACS-elements, and are integrated in the bit-slice implementation of the ACS-elements.
- Codes with different generator polynomials can easily be implemented by reprogramming of contacts/vias to the global BM-busses.
- Buffers between the global and the local channels reduce the maximal wire length.

Optimization of the global PM-channel involves:

1. Generating all possible rings.
   By exhaustive search we have found the following figures:

K	No. of DACS-elements	No. of rings
4	4	1
5	8	2
6	16	16
7	32	2048
8	64	unknown

2. Generating all possible channel routing tasks by cyclically shifting the rings one place at a time. Because of symmetry, 16 possible placements exist for each ring.
3. Solving all the generated tasks and selecting a solution with the smallest possible number of vertical tracks.

It should be noted, that the task of generating the rings is equivalent to the problem of finding "Hamilton Cycles" in directed graphs. However, we do not know of any general solution for the Viterbi decoder interconnection graph.

5 Implementation

Both the functional and the circuit design is rather straightforward:

- We use 7-bit Path Metrics, 4-bit Branch Metrics and renormalization at 64. Each ACS element contains a 9-bit register: 7 bits for the PM, 1 bit for the decision and 1 bit for an "initialization-flag". The operation of the ACS-block is controlled by a two bit function code:

Function	Description
Clear	Sets the contents of all registers to 0
Test	Connects the registers into a scan-path
Initialize	If "flag" then $PM = 64$
Normal	Normal ACS-operation

  A 7-bit Path Metric allows for extensions to rate $R = 1/3$ or $R = 1/4$ codes, but reduction to a 5-bit Path Metric is straightforward, because of the bit slice structure.

- The whole circuit is implemented using static circuitry, except for the global renormalization calculation. The adders and comparators use ripple carry propagation [10, figures 8.2, 8.4, 8.5]. This is a simple, but also fast solution, because the comparator starts calculating as soon as the least significant bits of the sums are calculated. The total delay equals 9 circuit stages from inputs to decision. A two-phase non overlapping clock is used (but the chips can also operate on a single clock phase and its complement).

A chip containing the described ACS-block as well as a 62 bit deep back-track block has been designed. The designs are currently being laid out using the MAGIC design-system [9] and NORCHIP's simplified design rules for a family of 2-micron CMOS N-well processes. The two blocks will be fabricated on separate chips during summer 1989, and the final chip containing both blocks is expected to be fabricated at the end of 1989.

A prototype chip containing a complete DACS-element including the local channel and the necessary environment for testing the renormalization has been fabricated via NORCHIP (november 1988) at European Silicon Structures Incorporated [11], [12]. A microphotograph of the DACS-element, for comparison with figure 5, is shown in figure 6. The chip works correctly and runs at speeds above 26 MHz under worst-case conditions ($V_{DD} = 4.75\ V$ and $T_A = 70\ ^0C$). This figure includes the delay in the local channel and in the buffers between the global- and the local channel.

A number of area measures for the ACS-block are listed below:

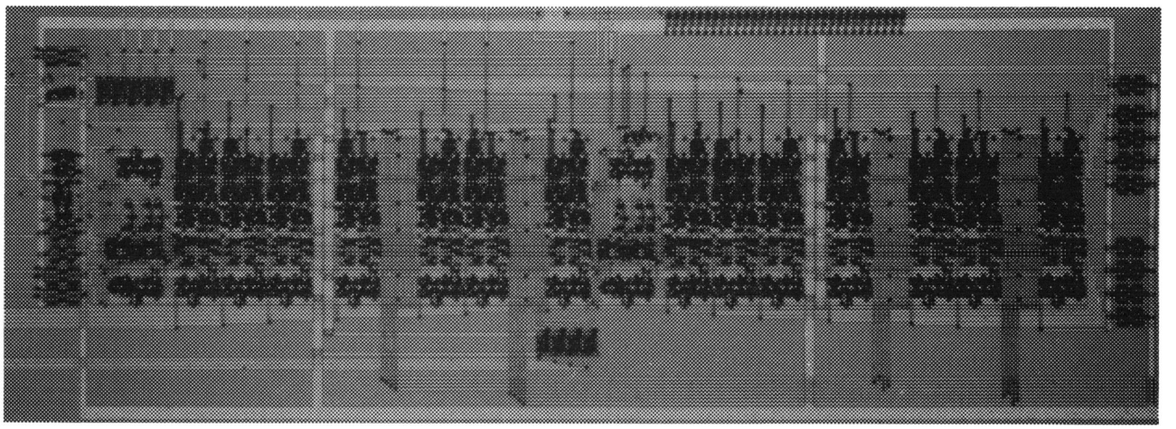


Figure 6: Microphotograph of the prototype DACS-element.

Dimensions:

	height (microns)	width
Complete ACS-block	7750	5097
Global PM-channel	7750	491
Column of 16 DACS-elements	7750	2303
DACS-element incl. local channel	483	2303
Local PM-channel	126	2167
Single ACS-element	357	1015

Percentage wiring area:

Global PM-channel	9.6 %
Local PM-channels	22.0 %
BM and power "slices"	10.5 %
Total	42.1 %

Transistor density:

The transistor density in the remaining area is 2100 transistors per mm^2

6 Conclusions

We have presented the floor-plan and implementation of the Add-Compare-Select processing elements of a fully parallel $K = 7, R = 1/2$ Viterbi decoder. The proposed two-column ring organization constitutes a regular layout-structure, in which the wiring area accounts for 42 % of the total area.

In addition, the structure described is suitable for codes with different values of K. This is because the global channel only takes up about 1/4 of the wiring-area (for $K = 7$), meaning that the wiring area is relatively insensitive to variations in K. The wiring can therefore be expected to take up about 40-45 % of the area of the complete ACS-block, for values of K ranging from $K = 5$ to $K = 8$.

The fabricated prototype DACS-chip operates at 26 MHz under worst case conditions. Taking into account the signal propagation delay in the global channel, speeds of 20-25 MHz are expected for the complete ACS-block.

References

[1] R. M. Orndoff et al, *Viterbi Decoder VLSI Integrated Circuit for Bit Error Correction,* Proceedings of 16th Annual Asilomar Conference on Circuits Systems and Computers, Nov. 8-10, 1982, pp. 149-152, IEEE Computer Society Press.

[2] J. B. Cain and R. A. Kriete, *A VLSI K=7, R=1/2 Viterbi Decoder,* Proc. of National Aerospace and Electronics Conference, Dayton, Ohio, 1984, pp. 20-27.

[3] P. G. Gulak, E. Shwedyk, *VLSI Structures for Viterbi Receivers: Part I General Theory and Applications,* IEEE Journal on Selected Areas in Communications, vol. SAC-4, no. 1, January 1986, pp. 142-154.

[4] P. G. Gulak, T. Kailath, *Locally Connected VLSI Architectures for the Viterbi Algorithm,* IEEE Journal on Selected Areas in Communications, vol. SAC-6, no. 3, April 1988, pp. 527-537.

[5] F. T. Leighton, G. L. Miller, *Optimal Layouts for Small Shuffle Exchange Graphs,* Proc. of VLSI 81, Ed. J. P. Gray, Academic Press 1981, pp. 289-299

[6] *Q-1401 Single-Chip Viterbi Decoder,* Qualcomm Incorporated, 10555 Sorrento Valley Road, San Diego, CA 92121.

[7] Consultative Committee for Space Data Systems, *Recommendation for Space Data System Standards, Telemetry Channel Coding,* CCSDS 101.0-B-2 Blue Book, January 1987.

[8] G. C. Clark, Jr. and J. B. Chain, *Error-Correcting Coding for Digital Communications,* Plenum Press, 1981.

[9] W. S. Scott, R. N. Mayo, G. Hamachi and J. K. Ousterhout, editors, *1986 VLSI Tools: Still More Works by the Original Artists,* Computer Science Division (EECS), University of California, Berkeley, ReportNo. UCB/CSD 86/272, December 1985.

[10] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design - A Systems Perspective,* Addison-Wesley, 1985

[11] J. Sparsø, E. Paaske, H. N. Jørgensen. *Introduction and Specification of a 10 Mbit/sek Viterbi dekoder chip (Viterbi Project-report no. 1),* Dept. of Computer Science, Dok. nr. ID-1225, Dept. of Telecommunication, Dok. nr. IT-8802, April 1988. (in danish).

[12] H. N. Jørgensen. *DACS-module test-chip (Viterbi Project-report no. 2),* Dept. of Computer Science, Dok. nr. ID-1226, Dept. of Telecommunication, Dok. nr. IT-8803, April 1988. (in danish).